

CLAIMS:

1. A value transaction system comprising a plurality of transaction units and a controller having a processor and memory means storing run-time interpreted code units each associated with a respective transaction unit, the controller being operable to execute the code of each respective code unit and in response thereto to generate signals controlling the operation of the respective transaction units.

2. A system as claimed in claim 1, further comprising a native code unit operable to accept and process input signals for the purpose of validation of an item of money.

a 3. A system as claimed in claim 1 or claim ~~2~~, wherein the transaction units are arranged to handle respective types of payment media.

4. A system as claimed in ^{*claim 1*}~~any preceding claim~~, wherein each interpreted code unit is independently functional without regard to the presence of the other interpreted code units.

5

6

10

- (b)

8

20

9

5 11. A transaction unit as claimed in ~~any one of claims 7 to 10~~,
wherein the transaction unit is a coin validation mechanism.

13. A transaction system as claimed in claim 12, wherein the transaction units are interconnected via a serial link.

the controller being operable to execute the code in each respective code module, the code in that module being functional independently of the

code in the other modules and performing processing operations in response to signals received from its respective transaction unit indicative of respective operations performed by that transaction unit, and the code being further operable to cause the controller to generate controlling signals for sending to the respective transaction unit and capable of representing different functions to be performed by the transaction unit.

15. A transaction system as claimed in claim 14, wherein the memory means has executable code in a further code module, that executable code being responsive to credit-representing signals generated by the code in one or more other code modules, and being operable to produce vend-authorising signals for use by the executable code in at least one other code module.

16. A transaction system as claimed in claim 14 ~~or claim 15~~, wherein the executable code is run-time interpreted code.

17. A transaction system as claimed in ~~any one of claims 14 to 16~~, wherein the controller is housed in one of the transaction units.

18. A transaction system as claimed in any one of claims ~~14 to 17~~^a, wherein each code module is contained in a respective area of protected memory.

5 19. A transaction system as claimed in any one of claims ~~14 to 18~~^a, wherein the executable code is Java bytecode.

20. A transaction system as claimed in any one of claims ~~14 to 19~~^a, wherein the transaction units are interconnected via a serial link.

10 21. A transaction system as claimed in any one of claims ~~14 to 20~~^a, wherein the transaction units include one or more of (a) a coin mechanism unit, (b) a banknote mechanism unit, (c) a card reader unit and (d) a vending machine controller unit.

15 22. A transaction system comprising a controller unit including a processor operable to execute instructions in Java code, and at least one transaction unit including means for performing value transactions under the control of the processor executing code uploaded from the transaction unit.

20 23. A transaction system as claimed in claim 22, wherein the transaction system comprises a plurality of transaction units, and the

controller unit is operable to execute code stored in respective code units each associated with a respective transaction unit.

24. A transaction system as claimed in claim 23, wherein the code
5 units are stored in respective protected memory areas.

25. A method of assembling a transaction system, the transaction
system comprising a plurality of transaction units and a controller having a
processor and memory means for storing executable code in respective code
10 modules each associated with a respective one of the transaction units, the
controller being coupled to the transaction units and arranged to receive and
send signals from and to the transaction units, and the controller being
operable to execute the code in each respective code module, each code
module performing processing operations in response to signals received from
15 the respective transaction unit indicative of respective operations performed
by that transaction unit, and the code module being further operable to cause
the controller to generate controlling signals for sending to the respective
transaction unit and capable of representing different functions to be
performed by the transaction unit; the method comprising:
20 separately loading the executable code for the respective code modules
into the memory means of the controller.